Fig. 1

Parsing system 100

Node generator 23

Grammar information object 24

Syntax recovery unit 22

Token recovery unit 12

Table 30

Identifier (ID) | Syntax recovery set

Parser 20
Context pointer
Buffer 21

Lexical analyzer 10
Buffer 11

Input stream

Input document

Abstract syntax tree (AST)

```
                        ┌─────────┐
                        │  Start  │
                        └─────────┘
                             │
                             ▼
S201 ──────────  ┌─────────────────────┐
                 │  Initialize  parser │
                 └─────────────────────┘
                             │
                             ▼
S202 ──────────  ┌─────────────────────┐        ┌──────────────────────────┐
                 │   Obtain  token  t  │        │ Generate node n and      │ ── S205
                 └─────────────────────┘        │ add  to  context  pointer│
                             │                  └──────────────────────────┘
                             ▼   S203                        │
         Yes      ╱───────────────────╲                      ▼
      ◄──────────╱   Is  token  t      ╲          ┌──────────────────────────┐
                 ╲     the  end ?      ╱          │ Change position of       │ ── S206
                  ╲───────────────────╱          │ context pointer to node n│
       ┌─────┐         No │    S204              └──────────────────────────┘
       │ End │            ▼                                  │
       └─────┘   ╱───────────────────╲    Yes                ▼         S207
                ╱   Does  token  t    ╲ ──────────►  ╱───────────────────╲
                ╲  match the grammar? ╱             ╱   Does  context     ╲    No
                 ╲───────────────────╱             ╲  pointer obtain all  ╱ ──────►
                      No │    S209                  ╲  the child nodes ?  ╱
                         ▼                           ╲───────────────────╱
                 ┌─────────────────────┐                      │ Yes
                 │ Report contents of  │                      ▼           S208
                 │       error         │             ┌──────────────────────┐
                 └─────────────────────┘             │ Change position      │
                         │                           │ of context pointer to│
                         ▼       S210                │ parent node          │
                ╱───────────────────╲    No          └──────────────────────┘
               ╱  Are unexecuted syntax╲ ─────────┐
               ╲  recovery units present?╱         │
                ╲───────────────────────╱          │
                      Yes │    S211                │
                          ▼                   S213 │
                 ┌─────────────────────┐           ▼
                 │ Execute first       │    ┌──────────────┐
                 │ unexecuted syntax   │    │ Default error│
                 │ recovery unit       │    │   process    │
                 └─────────────────────┘    └──────────────┘
                          │      S212
                          ▼
            No   ╱───────────────────╲
        ◄───────╱    Successful ?     ╲
                 ╲───────────────────╱
                      Yes │
```

Fig. 2

```
                    ┌──────────┐
                    │  Start   │
                    └──────────┘
                          │
                          ▼
                 ┌──────────────────┐
       S301      │   Initialize     │
                 │ lexical analyzer │
                 └──────────────────┘
                          │
                          ▼
                 ┌──────────────────┐
       S302      │ Obtain character │
                 │   string str     │
                 └──────────────────┘
                          │
                          ▼
                      S303
                    ╱◇◇◇◇◇╲          Yes
                 ╱ Can character string str ╲ ────────────┐
                ◇ be changed into token?    ◇             │
                 ╲                         ╱               ▼
                    ╲◇◇◇◇◇╱                          S304
                      No                    ┌──────────────────┐
                       │                    │  Transmit token  │
                       ▼                    │   to parser      │
                      S305                  └──────────────────┘
                    ╱◇◇◇◇◇╲          No                │
                 ╱  Are unexecuted syntax ╲ ──────────┐   │
                ◇  recovery units present? ◇          │   │
                 ╲                        ╱           │   │
                    ╲◇◇◇◇◇╱               │           │   │
                      Yes   S306          │           │   │
                       │                  │           │   │
                       ▼                  │    S308   │   │
                 ┌──────────────────┐     │           │   │
                 │ Execute first    │     ▼           │   │
                 │ unexecuted       │  ┌──────────┐   │   │
                 │ syntax recovery  │  │ Default  │   │   │
                 │ unit             │  │ error    │   │   │
                 └──────────────────┘  │ process  │   │   │
                       │               └──────────┘   │   │
                       ▼   S307              │         │   │
            No      ╱◇◇◇◇◇╲                  │         │   │
          ┌─────── ◇ Successful? ◇           │         │   │
          │         ╲◇◇◇◇◇╱                  │         │   │
          │            Yes                   │         │   │
          │             │                    ▼         │   │
          └─────────────┴────────────────────          │   │
```

Fig. 3

Analysis results for document A without error correction by syntax recovery unit
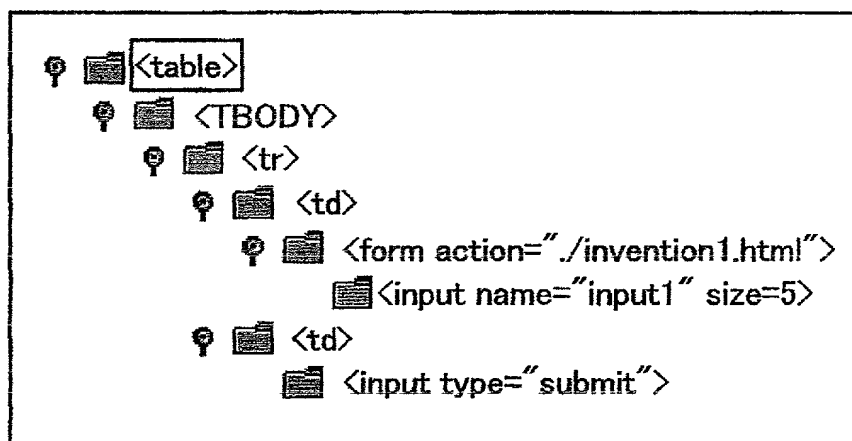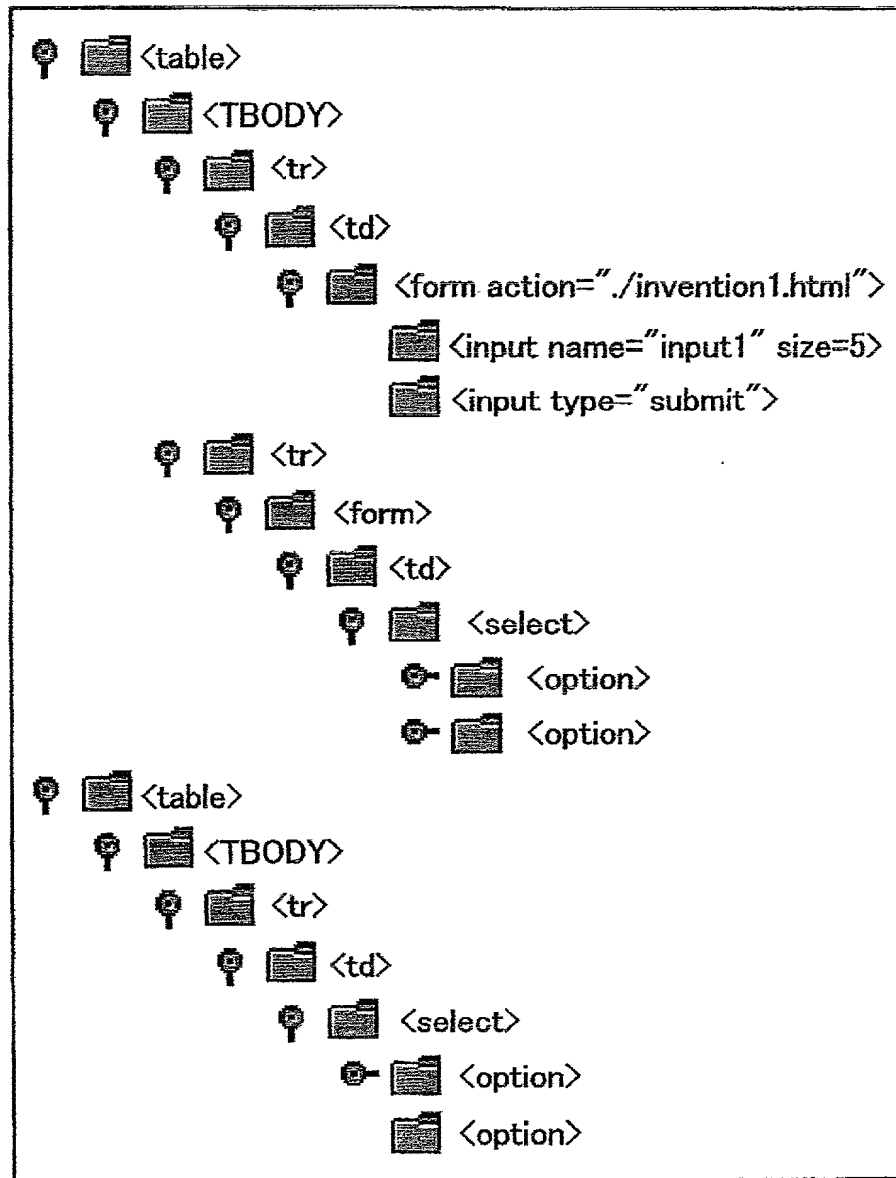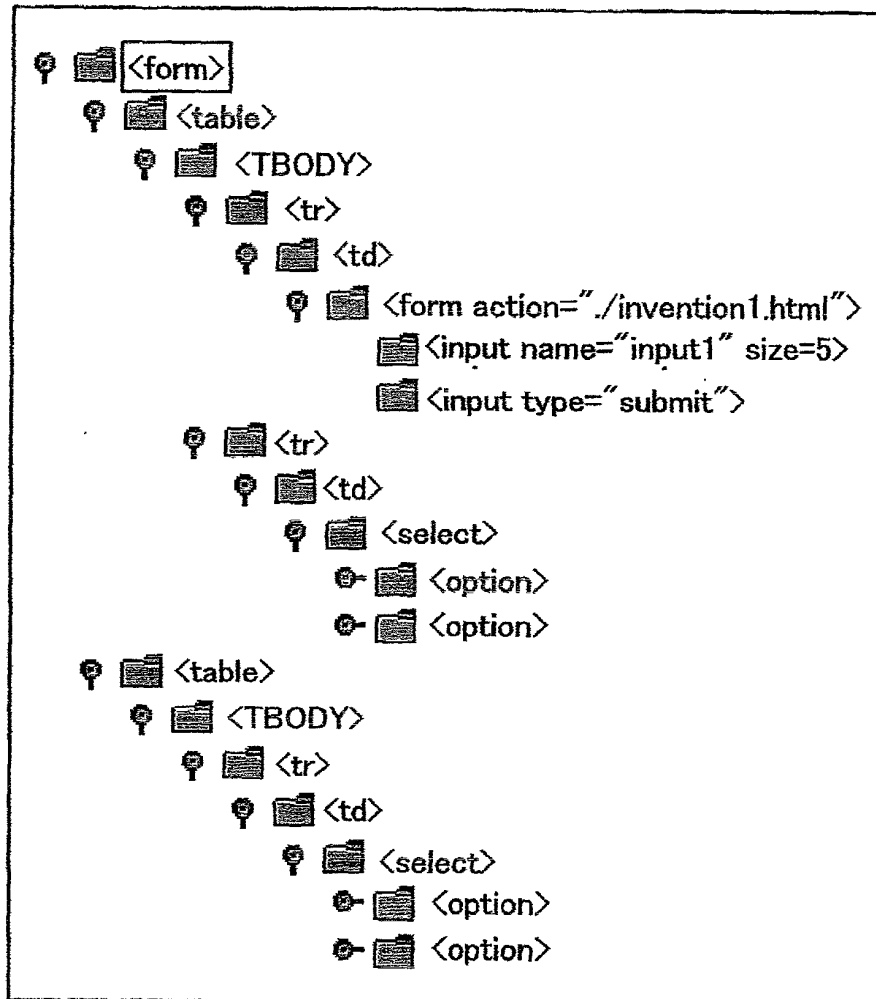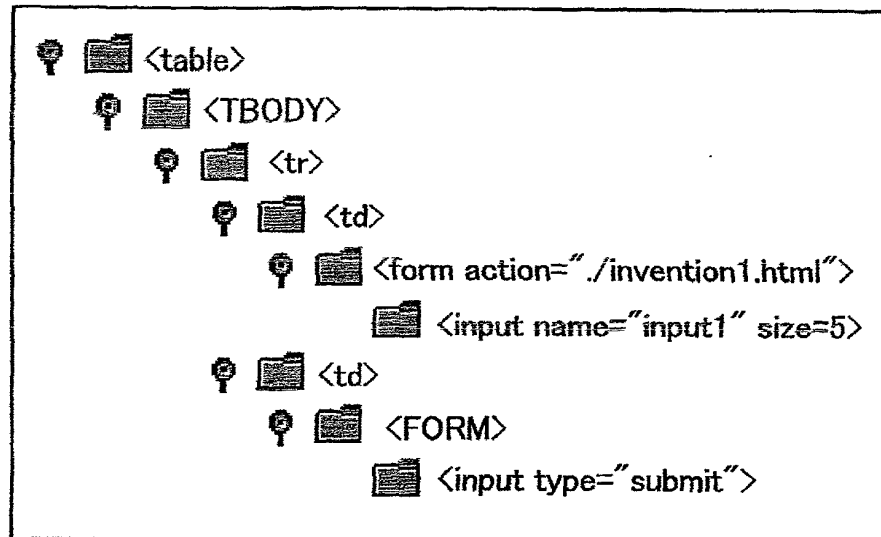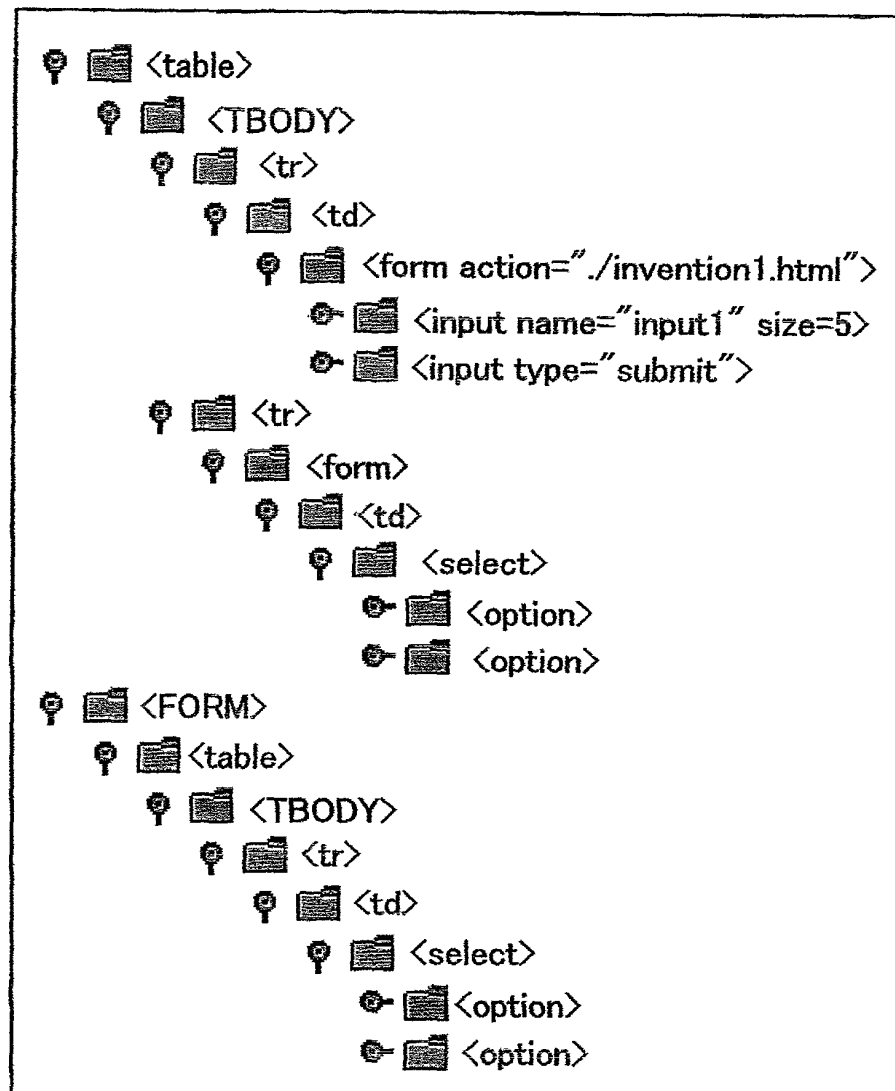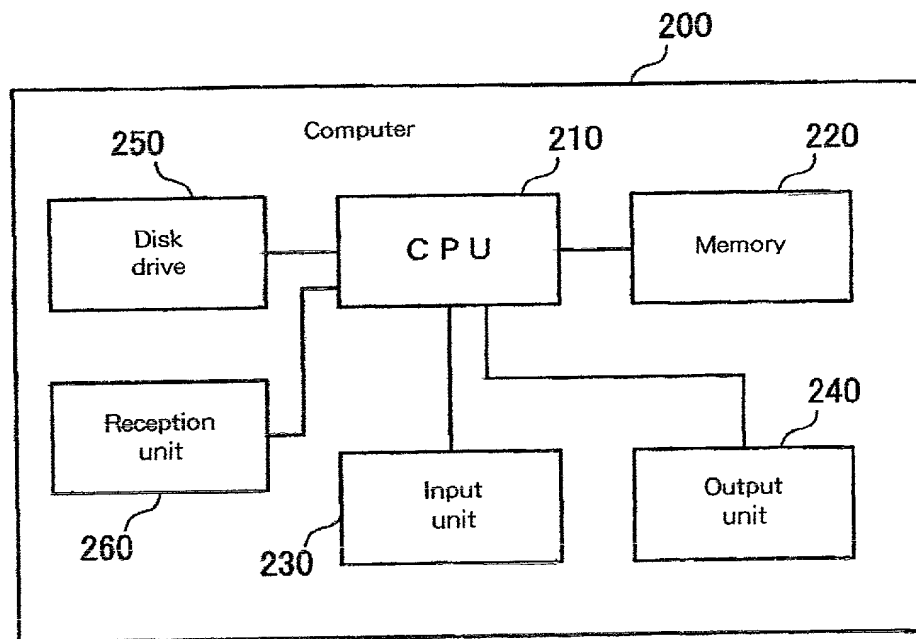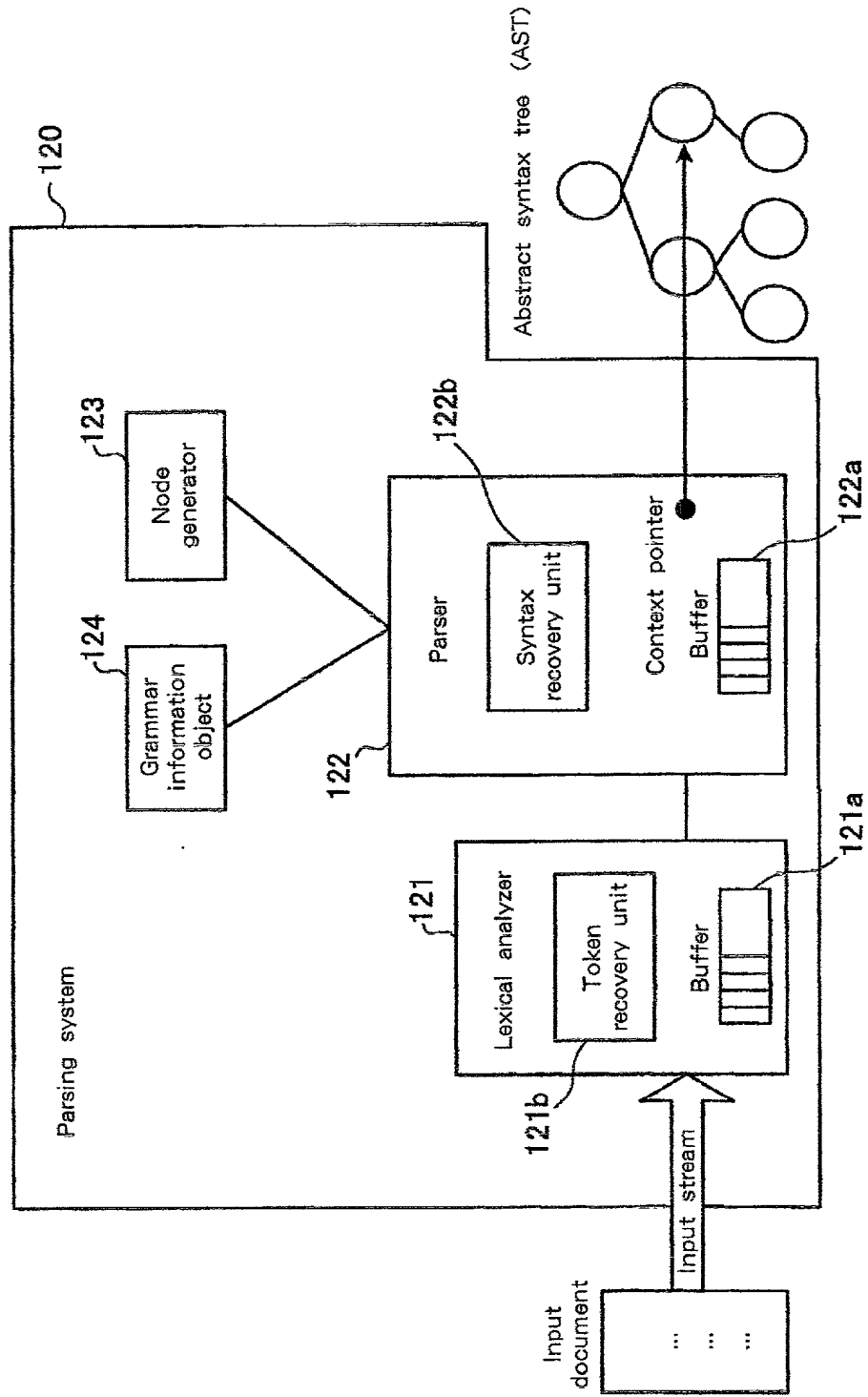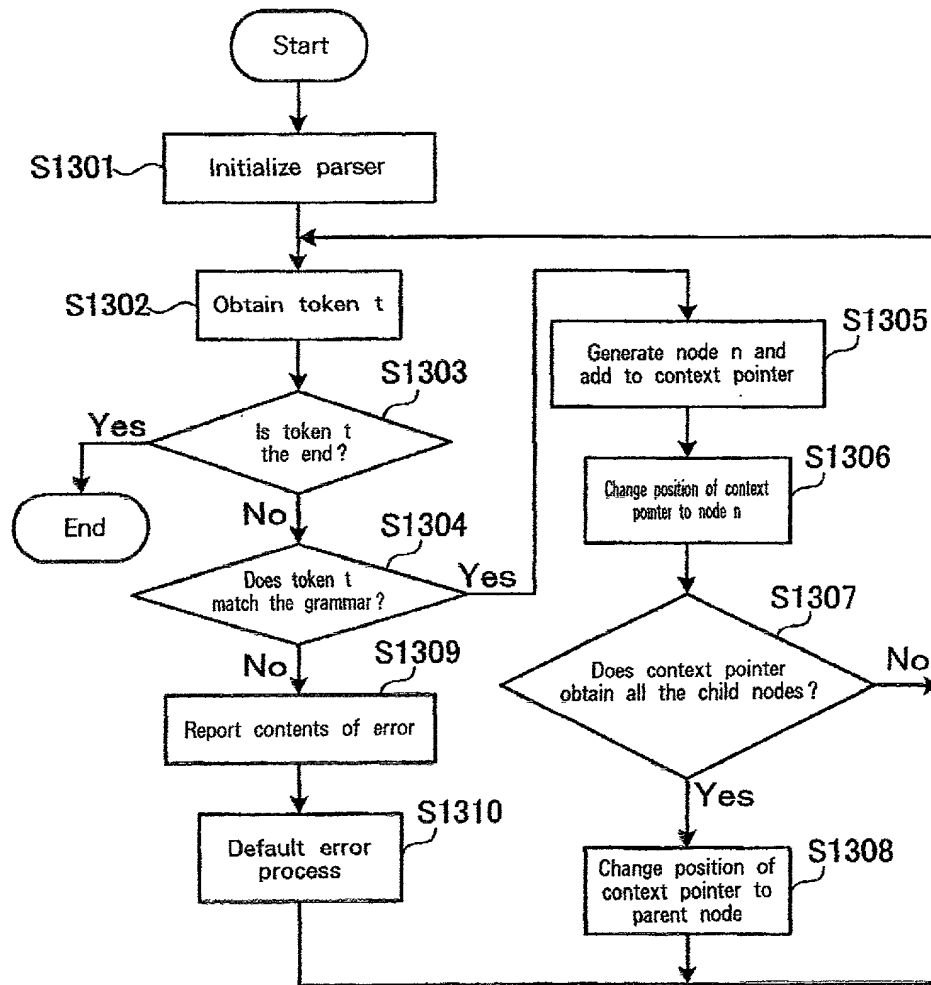
```
♀ 🖾 <table>
   ♀ 🖾 <TBODY>
      ♀ 🖾 <tr>
         ♀ 🖾 <td>
            ♀ 🖾 <form action="./invention1.html">
                  🖾 <input name="input1" size=5>
         ♀ 🖾 <td>
               🖾 <input type="submit">
```
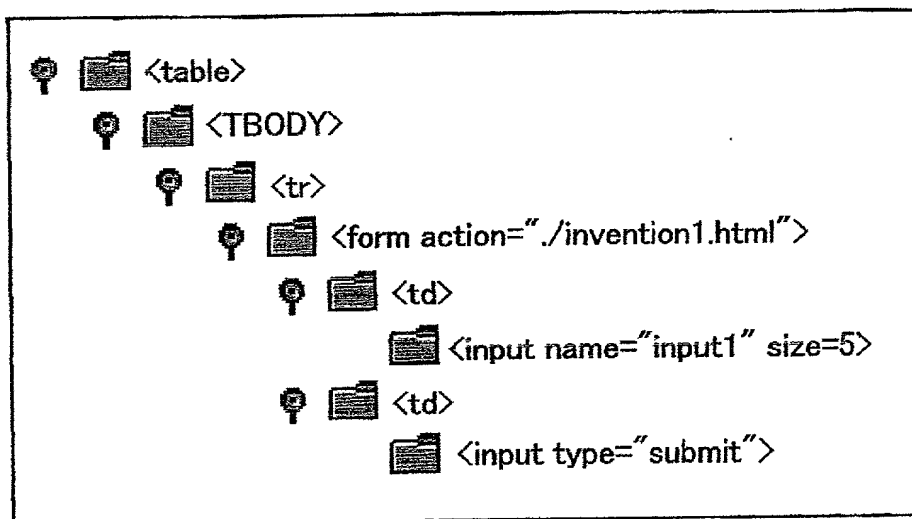
Fig. 4

Analysis results for document B without error correction by syntax recovery unit

```
📁 <table>
    📁 <TBODY>
        📁 <tr>
            📁 <td>
                📁 <form action="./invention1.html">
                    📁 <input name="input1" size=5>
                    📁 <input type="submit">
        📁 <tr>
            📁 <form>
                📁 <td>
                    📁 <select>
                        📁 <option>
                        📁 <option>
📁 <table>
    📁 <TBODY>
        📁 <tr>
            📁 <td>
                📁 <select>
                    📁 <option>
                    📁 <option>
```

Fig. 5

Analysis results for document A obtained using FormExpander


```
📁 <table>
    📁 <TBODY>
        📁 <tr>
            📁 <form action="./invention1.html">
                📁 <td>
                    📁 <input name="input1" size=5>
                📁 <td>
                    📁 <input type="submit">
```

Fig. 6

Analysis results for document B obtained using FormExpander

```
♀ 🖿 <form>
    ♀ 🖿 <table>
        ♀ 🖿 <TBODY>
            ♀ 🖿 <tr>
                ♀ 🖿 <td>
                    ♀ 🖿 <form action="./invention1.html">
                                🖿 <input name="input1" size=5>
                                🖿 <input type="submit">
                ♀ 🖿 <tr>
                    ♀ 🖿 <td>
                        ♀ 🖿 <select>
                            ●- 🖿 <option>
                            ●- 🖿 <option>
    ♀ 🖿 <table>
        ♀ 🖿 <TBODY>
            ♀ 🖿 <tr>
                ♀ 🖿 <td>
                    ♀ 🖿 <select>
                        ●- 🖿 <option>
                        ●- 🖿 <option>
```

Fig. 7

Analysis results for document A obtained using FormInserter


```
📁 <table>
   📁 <TBODY>
      📁 <tr>
         📁 <td>
            📁 <form action="./invention1.html">
               📁 <input name="input1" size=5>
         📁 <td>
            📁 <FORM>
               📁 <input type="submit">
```

Fig. 8

Analysis results for document B obtained using FormInserter

```
📍 🗃 <table>
   📍 🗃 <TBODY>
      📍 🗃 <tr>
         📍 🗃 <td>
            📍 🗃 <form action="./invention1.html">
               ⊙ 🗃 <input name="input1" size=5>
               ⊙ 🗃 <input type="submit">
         📍 🗃 <tr>
            📍 🗃 <form>
               📍 🗃 <td>
                  📍 🗃 <select>
                     ⊙ 🗃 <option>
                     ⊙ 🗃 <option>
📍 🗃 <FORM>
   📍 🗃 <table>
      📍 🗃 <TBODY>
         📍 🗃 <tr>
            📍 🗃 <td>
               📍 🗃 <select>
                  ⊙ 🗃 <option>
                  ⊙ 🗃 <option>
```

Fig. 9

30

Document identifier	Syntax recovery unit
Document A	[FormExpander]
Document B	[FormInserter]
Others	[]

Fig. 10

Fig. 11

TOSHIBA CORP 305080

Parsing system 120

Node generator 123

Grammar information object 124

Parser 122

Syntax recovery unit 122b

Context pointer

Buffer 122a

Lexical analyzer 121

Token recovery unit 121b

Buffer 121a

Input stream

Input document

Abstract syntax tree (AST)

Fig. 12

Fig. 13

str0 **str1** *str2* str3 str4

Fig. 14

str0 **str1** *str2* *str3* str4

Fig. 15